ELSEVIER
Available online at www.sciencedirect.com
SciVerse ScienceDirect
Procedia Engineering 29 (2012) 118 – 122
Procedia Engineering
www.elsevier.com/locate/procedia

2012 International Workshop on Information and Electronics Engineering (IWIEE)

The Complexity and Algorithm for Minimum Expense Spanning Trees

Zhan Ning[a], Wu Longshu[b*]

[a]*Mathematics and Computer Science Department, Xinyang Vocational and Technical College, Xinyang 464000, China*
[b]*College of Science, China Jiliang University, Hangzhou 310018, China*

Abstract

The minimum spanning tree problem is a classical and well-known combinatorial optimization problem. There exist many efficient algorithms such as the Kruskal algorithm and Prim algorithm to solve it. But in a real network, the vertices as well as the edges may have weights, and there are many cases of the vertex weights according to the degrees of the vertices. In this paper, we consider the computational complexity of the minimum expense spanning tree problem, which is to find a spanning tree in a network with minimum total expenses. We show that this problem is NP-hard in some general situations. And we propose a polynomial time algorithm when computing all the weights of the vertices in a spanning tree.

Keywords:Minimum expense spanning tree; Computational complexity; Polynomial time algorithm; Combinatorial optimization; NP-hard

1. Introduction

Compared with transmission networks, distribution networks has bigger scale and more possible compages of net configuration, so designing the configuration of distribution networks with computer aids is of great importance. This network must meet design planning requirements and minimum cost. Kruskal

* Corresponding author. Tel.: +0-86-571-86914424
E-mail address: wls@cjlu.edu.cn.

doi:10.1016/j.proeng.2011.12.679

algorithm and Prim algorithm are effective for the classic minimum spanning tree problems[1]. The minimum expense spanning tree problem can be widely used in the configuration of networks.

The theory of minimum expense spanning trees is developed from the cost of degree, distance, delay and so on. Usually, the cost of vertex exists independently. In this paper, we consider both of the edge weights and the vertex expenses in the minimum expense spinning tree problems.

Now we introduce the minimum expense spanning tree problem. We are given an undirected connected graph $G=(V,E)$. $w:E\to R^{+}$ is a function of edge lengths. $f:E\to R^{+}$ is a function of vertex expenses. Let T be any spanning tree of G . Denote the set of leaves by a finite set V_1, and the set of the intermediate vertices by V_2. Let $d(T,v)$ stand for the degree of vertex v in T.

In practice, the expenses of the vertices are closely related to the degrees of the vertices in T . So, we can obtain the total expenses $C(T)$ of a spanning tree T of G .

$$C(T)=\sum_{(v_i,v_j)\in T} w(v_i,v_j)+\sum_{v_k\in V} d(T,v_k)f(v_k). \tag{1}$$

Because the degree of a leaf is 1, we can change (1) as follows.

$$C(T)=\sum_{(v_i,v_j)\in T} w(v_i,v_j)+\sum_{v_k\in V_1} f(v_k)+\sum_{v_l\in V_2} d(T,v_l)f(v_l). \tag{2}$$

Therefore the problem is to find a spanning tree in a network with minimum total expenses.

2. Complexity Analysis and Algorithm Design

In this paper, we mainly consider the different cases of the expenses of vertices. We distinguish the following three situations.

2.1. We only consider the expenses of the leaves

We can change (2) as follows.

$$C(T)=\sum_{(v_i,v_j)\in T} w(v_i,v_j)+\sum_{v_k\in V_1} f(v_k). \tag{3}$$

We have the following Theorem 2.1.

Theorem 2.1 The minimum expense spanning tree problem is NP-hard, when we only consider the expenses of the leaves and the edge weights.

Proof: Because the problem to find a spanning tree in an undirected connected graph with maximum number of leaves is NP-hard[3]. We can polynomially reduct this problem to the minimum expense spanning tree problem.

For a given undirected connected graph $G=(V,E)$, $|V|=n$. We can construct the edge weight and vertex expense functions as follows. $w(e)=1,\forall e\in E$; $f(v)=1,\forall v\in V$.

Suppose T is a spanning tree with m leaves in G . That is $|V_1|=m$, $|V_2|=n-m$. So we can get:

$$C(T)=\sum_{(v_i,v_j)\in T} w(v_i,v_j)+\sum_{v_k\in V_1} f(v_k)=m+n-1 \tag{4}$$

We can get the conclusion that there is a spanning tree T with m leaves in G , if and only if there exists a minimum expense spanning tree with $C(T)=m+n-1$. Because the problem of the most leaves spanning tree is NP-hard, this completes the proof of Theorem 2.1.

2.2. We don't consider the expenses of the leaves

We can change (2) as follows.

$$C(T)=\sum_{(v_i,v_j)\in T} w(v_i,v_j)+\sum_{v_l\in V_2} d(T,v_l)f(v_l). \tag{5}$$

We distinguish the following two cases.

1. We ignore the degrees of the intermediate vertices. Then we can change (5) as follows.

$$C(T)=\sum_{(v_i,v_j)\in T} w(v_i,v_j)+\sum_{v_l\in V_2} f(v_l). \tag{6}$$

We have

Theorem 2.2 The minimum expense spanning tree problem is NP-hard, when we only consider the expenses of the intermediate vertices (with no degree) and the edge weights.

Proof: For any given undirected connected graph $G=(V,E)$, $|V|=n$. We can construct the edge weight and vertex expense functions as follows. $w(e)=1,\forall e\in E$; $f(v)=1,\forall v\in V$.

Suppose T is a spanning tree with m leaves in G. That is $|V_1|=m$, $|V_2|=n-m$. We have

$$\begin{aligned}C(T)&=\sum_{(v_i,v_j)\in T} w(v_i,v_j)+\sum_{v_l\in V_2} f(v_l)\\&=n-1+n-m\\&=2n-m-1.\end{aligned} \tag{7}$$

We can get the conclusion that there is a spanning tree T with at least m leaves in G, if and only if there exists a minimum expense spanning tree with $C(T)\le 2n-m-1$. Because the problem of the most leaves spanning tree is NP-hard, Theorem 2.2 is proved.

2. We consider the degrees of the intermediate vertices. We can change (5) as follows.

$$C(T)=\sum_{(v_i,v_j)\in T} w(v_i,v_j)+\sum_{v_l\in V_2} d(T,v_l)f(v_l). \tag{8}$$

We have

Theorem 2.3 The minimum expense spanning tree problem is NP-hard, when we only consider the expenses of the intermediate vertices (with degrees) and the edge weights.

Proof: Given any undirected connected graph $G=(V,E)$, $|V|=n$. We can construct the edge weight and vertex expense functions as follows. $w(e)=1,\forall e\in E$; $f(v)=1,\forall v\in V$.

Suppose T is a spanning tree with m leaves in G. That is $|V_1|=m$, $|V_2|=n-m$. Because $d(T,v)\le m$, $\forall v\in T$. We have

$$\begin{aligned}C(T)&=\sum_{(v_i,v_j)\in T} w(v_i,v_j)+\sum_{v_l\in V_2} d(T,v_l)f(v_l)\\&\le n-1+m(n-m)\\&=n+mn-m^2-1.\end{aligned} \tag{9}$$

We can get the conclusion that there is a spanning tree T with m leaves in G, if and only if there exists a minimum expense spanning tree with $C(T) \le n + mn - m^2 - 1$. Because the problem of the most leaves spanning tree is NP-hard, the proof of Theorem 2.3 is completed.

2.3. We consider the expenses of all the vertices

We can distinguish the following two cases.

1. $f(v_i) = k$ ($k \in Z^+$ is a constant, $1 \le i \le n$), we can change (1) as follows.

$$C(T) = \sum_{(v_i, v_j) \in T} w(v_i, v_j) + \sum_{v_k \in V} d(T, v_k)k. \tag{10}$$

For an undirected connected graph $G = (V, E)$, $|V| = n$. We can get $\sum_{i=1}^{n} d(v_i) = 2|E|$. For a spanning tree $T = (V, E_1)$, we have $|E_1| = |V| - 1 = n - 1$. So we can get

$$C(T) = \sum_{(v_i, v_j) \in T} w(v_i, v_j) + 2k(n-1). \tag{11}$$

Then we can use Kruskal algorithm or Prim algorithm to solve it. We can obtain the minimum spanning tree only with the edge weights. Suppose the optimal value is OPT. So we can obtain the optimal value $C(T)_{OPT}$ of the minimum expense spanning tree problem.

$$C(T)_{OPT} = OPT + 2k(n-1). \tag{12}$$

2. If $f(v_i)$ ($1 \le i \le n$) is not a constant. Then we can change (1) as follows.

$$C(T) = \sum_{(v_i, v_j) \in T} (w(v_i, v_j) + f(v_i) + f(v_j)). \tag{13}$$

For this kind of optimization problem, we can let $w'(v_i, v_j) = w(v_i, v_j) + f(v_i) + f(v_j)$ for each edge $e = (v_i, v_j) \in E$, then by using Kruskal or Prim algorithm, we can obtain the minimum spanning tree with edge weight $w'(v_i, v_j)$. The optimal value is $C(T)_{OPT}$. We present the algorithm as follows:

Algorithm 2.1 (G, w')

$A \leftarrow \varnothing$
for each vertex $v \in V$
    do create a new set whose only member is v
sort the edges of E into nondecreasing order by weight w'
for each edge $(u, v) \in E$, taken in nondecreasing order by weight
    do if Find-Set $(u) \neq$ Find-Set (v)
        then $A \leftarrow A \cup \{(u, v)\}$
            unite the dynamic sets that contain u and v into a new set that is the union of these two sets
return A

where Find-Set (x) returns a pointer to the representative of the unique set containing x.

3. Conclusion

In this paper, we analysis several kinds of the minimum expense spanning tree problems and obtain many properties. Finally, we present the polynomial time algorithm for the minimum expense spanning tree problem. These properties and algorithms may apply to network planning and construction[2,4,5,6]. Moreover, for NP-hard problems, we can further research the design of approximation algorithms.

Acknowledgements

The authors are grateful to the supports of National Natural Science Foundation of China (No. 11171316), Zhejiang Provincial Natural Science Foundation of China (No. Y6090472) and the Scientific Research Fund of Zhejiang Provincial Education Department (No. Y201018835).

References

[1] Bondy JA, Murty USR. *Graph Theory with Applications*. The Macmillan Press; 1976.

[2] Goto H, Hasegawa Y, Tanaka M. Efficient Scheduling Focusing on the Duality of MPL Representatives. *Proc. IEEE Symp. Computational Intelligence in Scheduling* (*SCIS 07*) 2007; 57-64.

[3] Grey MR, Johnson DS. *Computers and Ibractabilit: A Guide to the Theory of NP-completeness*. San Francisco: Freeman; 1978.

[4] Jacobs IS, Bean CP. *Fine particles, thin films and exchange anisotropy*, in Magnetism, vol. III. New York: Academic; 1963, p. 271–350.

[5] Kornack D, Rakic P. Cell. Proliferation without Neurogenesis in Adult Primate Neocortex, *Science* 2001; 294: 2127-2130.

[6] Yorozu Y, Hirano M, Oka K, Tagawa Y. Electron spectroscopy studies on magneto-optical media and plastic substrate interface. *IEEE Transl. J. Magn. Japan* 1987; **2**: 740–741.